Vorticity Confinement Modeling of Dynamic Stall With Tight Structural Coupling§

John Osburn Bridgeman
BEER
2360 Venn Ave.
San Jose, CA 95124-4927

and

William E. Dietz
Flow Analysis, Inc.
Tullahoma, TN 37330

Abstract

Dynamic stall is an aerodynamic phenomenon encountered when rotorcraft operate near performance boundaries typical of high-speed or maneuvering flight. The complex, highly non-linear aerodynamics and associated interaction with the structures induce exceptionally high loads, violent vibrations, reduced handling qualities, and a reduction in fatigue life. The prediction of dynamic stall remains one of the most challenging tasks for engineers, necessitating a tightly coupled aerodynamic/structures predictive capability. Comprehensive models are efficient but are generally not valid for a wide range of geometries. Conventional Navier-Stokes approaches have shown promise but their computational cost is prohibitive. Flow Analysis, Inc. (FAI) has developed a technique for predicting dynamic stall using the Vorticity Confinement method incorporated into a compressible flow solver. This approach requires far fewer computational resources when compared to conventional methods. The Vorticity Confinement method has been implemented in the compressible Euler/Navier-Stokes rotor code, TURNS_SERIAL, and used to predict dynamic stall events for the NACA 0015 airfoil, including comparisons with experimental data. A two-degree-of-freedom structural dynamics model has been coupled to the flow solver and stable computations have been demonstrated. A parallel implementation of the flow solver, TURNS_MPI, has been used to demonstrate scalability on a parallel processor for both two- and three-dimensional steady computations.

Introduction

Rotary-wing aircraft routinely operate in a very complex aerodynamic environment that limits their performance and provides a wide variety of significant and challenging problems for engineers. The most critical problems that factor into the design process are those associated with the main rotor system. Severe and potentially dangerous aerodynamic environments can be encountered in operating conditions that push the edges of the performance envelope, such as very high speed and maneuvering flight. In addition to the strong induced flow effects and interactions with the wake in routine operation, the rotor can experience the effects of transonic flow, flow reversal, and dynamic stall. Cases have been reported where limit loads were associated with strong unsteady, compressible and three-

Presented at the *American Helicopter Society* Aerodynamics, Acoustics, and Test and Evaluation Specialists Meeting, San Francisco, California, January 23-25, 2002.

§Research supported by DoD SBIR Phase I Contract Number DAAH10-01-C-0009.

dimensional flow at the blade tip not attributed to cyclic stall events [1]. However, the real limiting factor is associated with the flow on the retreating side [2] where, in an effort to balance loads across the rotor disk, very high angles-of-attack are required and dynamic stall is encountered. These conditions produce exceptionally large torsional moments, excessive vibratory and torque loading, a deterioration in handling qualities, and ultimately fatigue of major components. The design criteria associated with these events are sufficiently influenced that they dictate the sizing of control system components [3]. Understanding of the phenomenon is requisite to its prediction.

Dynamic stall is an aerodynamic flow condition that results from the forced time-dependent pitching (or plunging, flapping, etc) motion of an aerodynamic surface producing a cyclic event with large separation and subsequent reattachment of the flow, resulting in a large hysteresis in the loading on the vehicle. The angle of incidence at which stall occurs is well beyond the angle for static stall and is described as the dynamic stall delay angle. The separation can initiate at either the leading or trailing edge and ultimately results in the release of a concentrated vortex from the leading edge (l.e.) region which convects over the blade at significantly less than (approximately 30%) the free-stream velocity [4]. Corresponding to the shedding of the l.e. vortex is a sudden and dramatic increase in nose-down pitching moment, typically referred to as *moment stall*. The vortex convection over the surface enables a continued lift increase, until it passes the trailing edge (t.e.), at which point the lift abruptly decreases well below the static value, and is referred to as *lift stall*. McCroskey, et al [5] have separated the dynamic stall phenomenon into four distinct categories: (1) No stall (attached flow); (2) Stall onset; (3) Light stall; and (4) Deep stall. In the limiting case of deep stall the shed vortex is well defined and the event less sensitive to the parameters that characterize the geometry and flow, but the transient loads can be several times static values. However, in the less severe conditions of stall onset through light stall, there is an extreme sensitivity to airfoil shape, mean and oscillatory amplitude of the motion, reduced frequency, and Mach number. In light stall the loads are typically close to their static values, but the greatest negative aerodynamic damping is produced which can lead to stall flutter.

Helicopters experience dynamic stall when the rotors are forced to operate close to their thrust limit, such as in maneuvers. Under these conditions, the large nose-down pitching moments, coupled with an inherent softness in the control system, lead to large blade torsional motions, which in turn affect the aerodynamics. Bousman [6] identified a number of stall events from data taken in the UH-60A Airloads Program. The major conclusion from the investigation was that torsional dynamics indicated *where* dynamic stall might occur and the associated aerodynamic inflow determines *whether* it will occur. Clearly the coupled response of the system suggests this problem should be treated in a multidisciplinary manner.

The ability to accurately predict dynamic stall continues to be one of the greatest challenges to the rotorcraft community. Considerable insight has been gained through extensive wind tunnel testing from which researchers have been able to construct efficient empirical models that are readily incorporated into comprehensive codes [7,8]. Unfortunately, they are inherently two-dimensional and highly tuned to a particular airfoil test. Given the extreme sensitivity of stall conditions it's not surprising that they perform poorly in general application. Empiricism can be avoided by using first-principles-based methods for dynamic stall prediction, and should capture the essential features of the flow in dynamic stall events, viz. flow separation dynamics and vortex convection. Computational Fluid Dynamics (CFD) methods, using the Reynolds-averaged, thin-layer Navier-Stokes equations have been applied to the prediction of dynamic stall on airfoils with some success and show promise. However, the persistent problem and central topic of discussion are the turbulence models. In addition to the general lack of consistent results are the prohibitive computational costs. Typically fine grids and small time steps are required, particularly for moment resolution, resulting in computing times of several hours per cycle of oscillation on a Cray C90 [9]. In fact, even in attached flow it may be necessary to use 10000 time steps per cycle of oscillation for accurate moments [10]. The implication for rotorcraft use is hundreds of hours of C90 time for a maneuver case. It's worth noting that for Reynolds numbers typical for rotor flows, the normal spacings at the surface, $\Delta\eta \cong 10^{-5}$ chords, are very fine and thus the majority of the CPU time is spent within a few

percent of chord from the surface, in an attempt to resolve the fine details of the turbulent boundary layer. However, that may not be necessary.

Flow Analysis, Inc. (FAI) has developed a technique for predicting dynamic stall that is based on their Vorticity Confinement method. Any suitable flow solver, when modified with vorticity confinement, is capable of efficiently and accurately predicting stall using coarse inviscid-type grids. Moulton [11] recently demonstrated this method was capable of accurately predicting several stall conditions on different airfoils in 2 ½ minutes of CPU time per cycle of oscillation on the C90. Given that a severe maneuver case may require many revolutions for a periodic solution, and the typical two- to three-dimensional increase in grid points, an accurate solution should cost a few hours on a vector machine such as the Cray.

Rotorcraft CFD research has been an intense and long term effort. But by-and-large, there has been little or no impact on the more routine comprehensive design and analysis process. This could be due to conservative design philosophies, computational cost, or an insufficient experience base for confidence. Even in the CFD/comprehensive couplings to date, there has been no conclusive demonstration of superior results. It's possible that inadequate coupling, insufficient aerodynamic fidelity, or application to cases where the flow conditions were too benign may be responsible. Perhaps an application of a very high fidelity aerodynamics method, in a tight coupling scheme, applied to a severe condition, might lead to the most effective demonstration of the improvements that can be gained from CFD.

There have been several attempts to capitalize on the higher fidelity aerodynamics from CFD methods. In a pioneering effort, Tung et al [12] coupled the FDR finite difference code, based on the transonic small-disturbance equation, to CAMRAD by using the CFD prediction for the lift in place of airfoil tables in a region on the advancing side. The trim procedure used by CAMRAD remained unchanged. After a converged comprehensive solution was obtained, the relevant wake and structural deformation information were combined into a partial inflow angle which, neglecting the wake represented by the CFD code, was used as a blade surface boundary condition. The entire procedure converged in a few cycles. The individual disciplines could only affect one another outside the trim procedure. This approach has generally been described as *loose* coupling. The aerodynamics model was improved by using a full-potential formulation. The FPR [13] code, which developed from the TUNA (Transonic UNsteady Aerodynamics) code of Bridgeman, et al, [14] was used in the same coupling procedure, but the CFD lift was used over the entire rotor disc and the grid represented only the outer 50% of radius of the blade. Subsequent application of the coupled procedure to the Puma blade [15] produced no noticeable improvement over the comprehensive code. Strawn and Bridgeman [16] improved their model by including pitch rate and lead-lag information, and attempted to couple the moments. However the solution procedure was unstable. It was speculated that the loose coupling procedure was insufficient to resolve the interaction between the pitching motion and the torsional dynamics. The most aggressive coupling to date between CFD and a comprehensive code is the development by Lee, et al [17]. The FPX code of Bridgeman, et al. [18] was tightly coupled to 2GCHAS [19] using all forces and moments and full-range motion of the CFD grid using structural deformation information. Initial solutions were unstable due to the presence of erroneous negative aerodynamic damping from the CFD moments. Ultimately, a temporal resolution was required for moments that was *4* times finer than needed for lift ($\Delta\psi \cong ¼^{o}$), which appears to consistent with other CFD codes. Subsequent computations for the Puma case were stable and accurate. Unfortunately the implication was a need for substantial computational resources, even for the full-potential formulation.

Most flow solvers of interest execute very efficiently on vector mainframes. However these computers are not generally accessible to the rotorcraft community. Assuming current trends in computer resources, these machines will be virtually inaccessible soon and are certainly not the best environment in which to perform multidisciplinary computations anyway. Finally, they typically have a limited number of processors and cannot provide any reasonable scalability. An effective use of parallel computing resources is mandatory for these problems, especially considering the frequent disparate computational intensity of individual disciplines. Significant to the private sector is the fact that they are generally

available not just as mainframe systems, but also in the form of NoWs (Networks of Workstations) and even Beowulf clusters.

Unfortunately, most of these codes do not take advantage of parallel computing resources. This is principally due to a more substantial investment in conversion than that required by vectorization. Recent improvements in hardware, languages, compilers, auto-parallelizing software, and parallelizing standards have made the conversion tasks far easier. Fortran 90 and 95 have embedded parallelism, and HPF (High Performance Fortran) provides additional improvement. The OpenMP standard uses simple directives placed in appropriate portions of the code to aid the compilers to parallelize the code. It has the significant advantage that code conversion can be completed in a matter of days with no effect on its performance on other machines (e.g. the Fpx code was parallelized in 2 days including the requisite education). These are techniques that are based on the data parallel programming concept, which is acceptable for shared memory systems. However they are not effective on distributed memory systems, do not provide a standard for running different disciplines in parallel, and do not scale well for a large number of processors. One point to be made is that true scalability typically requires scaling the problem size with the number of processors. Otherwise, there is a point of diminishing returns that is associated with Amdahl's law and the percentage of run time associated with inter-processor communications.

Presently, use of the MPI (message passing interface) standard is the most effective approach for both shared and distributed memory systems (although combined MPI/OpenMP constructs for SMPs - symmetric multi-processors - is gaining attention), can guarantee parallelism at the individual and multiple discipline levels, is highly portable, and problems scale nearly linearly. The major disadvantage is that the code conversion process is substantial and can cause changes in its ability to execute effectively on serial machines. For the aerodynamics discipline, the paradigm is domain decomposition with MPI for communications and at the multiple discipline level, concurrent execution of structural dynamics and aerodynamics codes is accomplished using MPIRUN. These facts clearly indicate the optimum scenario is to obtain an aerodynamics code with an *existing* MPI construction to minimize on development time and have optimum parallel capability.

Our views on the most important elements of this effort are described below. The principle force that drives the success of the work as defined in present topic is clearly the aerodynamics. State-of-the-art comprehensive codes have a long history of development using very efficient models for the entire aeromechanics domain. Their computational requirements will be a small fraction of the total computational cost. The requirement for a first-principles-based method cannot be met with practicality using present Navier-Stokes solvers with turbulence modeling. Some form of unique model must be used other than direct PDE resolution of the boundary layer for fast computations (relative to typical costs of high-fidelity aerodynamics). The design of the CSD/CFD coupling procedure must start with a parallel computing capability and have enough flexibility to shift computational tasks between disciplines to guarantee proper load balancing. The level of effort of the aerodynamics model must be able to be increased in response to the problem, such as computing the solution over the rotor disk using an isolated blade, or all of the blades simultaneously, as may be required for maneuvering flight.

The fundamental goal of the present effort is to demonstrate the feasibility of developing a rotorcraft comprehensive design and analysis code with superior accuracy, applicability, and reliability in predicting performance in extreme flight conditions, such as maneuver, which can experience dynamic stall events. Such a tool may be realizable through the integration of several existing and openly available first-principles-based aerodynamic codes and comprehensive rotorcraft tools, FAI's unique dynamic stall prediction method, and a plan for judicious use of parallel processing resources. The tasks defined to accomplish this are: (1) incorporate the vorticity confinement method into a compressible flow solver with previously demonstrated success in application to rotor blades and preferably with a parallel construction, and validate through comparisons with experimental data; (2) tightly couple the aerodynamics code and a simple structural dynamics code using a time-domain solution procedure; and (3) demonstrate the

scalability of the flow solver on a representative parallel architecture.

The compressible flow equations and chosen solvers, vorticity confinement method and implementation strategy, and structural dynamics equations and solution procedure are reviewed below. Computed results for a NACA 0015 airfoil in attached flow, light, and deep stall with comparisons to experimental data are then presented, along with coupled aero/structures solutions, and scalability characteristics. Finally, some conclusions and anticipated benefits of the present work for achieving the long-term goal are offered.

Mathematical Formulation

The principle mathematical models and associated codes used in this work are described below. The compressible flow equations are only briefly described as they are well documented. The serial and parallel flow solvers chosen are described in some detail, particularly, the differences between the two versions of the solvers that may impact the success of the long range goal. A discussion of the vorticity confinement procedure follows the traditional incompressible flow implementation for simplicity and consistency with our previous reports in the literature and because it succinctly describes the salient features. Its implementation in the serial flow solver is also discussed. Finally, the two-degree-of-freedom structural dynamics model, classically defined in textbooks, is offered here for completeness and to facilitate a discussion of some creative ideas that may be required to fully accomplish the multidisciplinary parallelism in the long term.

Compressible Flow Equations

The fluid physics associated with the operating environment of rotorcraft are generally well represented by the Reynolds-averaged Navier-Stokes (RANS) equations. The formulation is applicable to tip vortex and wake formation, convection, and interaction, strong compressible flow effects, and is suitable for most of the viscous flow phenomena. In turbulent flows, the influence of the fluctuating terms on the mean flow is accounted for through an eddy viscosity term provided by a suitable turbulence model. However for flows with strong separation effects, the validity of the predictions is questionable and certainly a function of that turbulence model. Numerical methods for the solution of these equations typically use the *thin-layer* approximation that viscous stresses in the stream-wise and cross-flow directions are negligible. When using a coordinate transformation from Cartesian coordinates *(x,y,z,t)* to a body-conforming system (ξ,η,ζ,τ) they are expressed as

$$\frac{\partial \boldsymbol{Q}}{\partial \tau}+\frac{\partial \boldsymbol{E}}{\partial \xi}+\frac{\partial \boldsymbol{F}}{\partial \eta}+\frac{\partial \boldsymbol{G}}{\partial \zeta}=\frac{1}{Re}\frac{\partial \boldsymbol{G_V}}{\partial \zeta}+\frac{\rho}{J}(\boldsymbol{R}+\boldsymbol{B}) \qquad (1)$$

where $\boldsymbol{J}$ is the Jacobian of the transformation, $\boldsymbol{Re}$ is the Reynold's number; $\boldsymbol{JQ}$ is the vector of conserved variables: mass, momentum, and energy; $\boldsymbol{E, F, G}$ are the inviscid flux vectors, $\boldsymbol{G_V}$ is the viscous flux vector, and $\boldsymbol{B}$ is a vector representing body force terms, which is typically zero for rotorcraft applications. For flow solvers using a non-inertial system, $\boldsymbol{R}$ is the source term accounting for the centrifugal acceleration of the rotating blades.

Compressible Flow Solvers

There are numerous approaches to solving these equations and the associated computer codes. However, it is important to choose a solver that satisfies the constraints as defined in the objectives, is mature and openly available (preferably one with which there has been prior experience), and, if possible, has an existing parallel capability. At present, there is only one code that satisfies all of the above.

TURNS Code

One of the most widely exercised compressible flow codes in the rotorcraft industry has been TURNS (Transonic Unsteady Rotor Navier-Stokes) **[20].** The code predicts the flow about an isolated rotor on a structured C-H topology grid, using an implicit finite difference method, which is well documented in the literature. In the solution procedure, these equations are discretized on a structured, blade-fixed grid and solved using an implicit, finite-difference scheme. The inviscid fluxes on the right-hand-side are computed using Roe's upwind biasing in all three directions, which eliminates the need for the explicit addition of

numerical dissipation, used in central-difference-based schemes, thereby producing a less dissipative scheme. A high-order MUSCL-type limiter is also used on the right-hand-side to produce a TVD (total variation diminishing) scheme, which improves the spatial accuracy to second- or third-order. The Baldwin-Lomax algebraic model is used for turbulent flows. An LU-SGS (Lower-Upper Symmetric Gauss Seidel) implicit operator is used (for improved stability and accuracy) to advance the equations at each step and the quasi-Newton method is used for sub-iteration at each step. This code has been routinely used on vector computers for over a decade. For the remainder of this paper, this serial/vector code will be referred to as TURNS_SERIAL to distinguish it from the parallel version discussed below. This is a mature code, no longer under development and thus stable, less restricted than current government codes (which may allow for commercialization), is used by many rotorcraft researchers, and FAI has ten years of experience with the code. It is optimum for the dynamic stall development, validation, and coupled structures computations. However, as it is a serial/vector code, it will not be suitable for the ultimate goal of scalable parallel processing.

TURNS_MPI Code

A version of this code has been modified for scalable performance on parallel machines, which we will refer to as TURNS_MPI [21] . The TURNS_MPI code uses a domain decomposition approach which guarantees a proper load balance by distributing an equal number of grid points to each of the processors. MPI (message passing interface) calls are used for inter-processor communication. Generally, the particulars of the algorithm and solution procedure are unchanged with respect to the serial/vector code, with two exceptions, which must be carefully investigated within to ensure the accuracy of the predictions. First, the LU-SGS implicit scheme is hybridized by passing processor boundary information to other processors at the start of the sweeps, and then only applied on each processor. This may have some effect on the time accuracy of the code (although no problems have been reported) when applied to highly transient flows. The second difference that must be investigated concerns the partitioning strategy. In TURNS_MPI, column-type decomposition is used where the segmented grids extend normally from the blade surface to the outer boundary to simplify boundary condition implementation. However, that implies that there are locations along the stream-wise and radial direction that will be inter-processor boundaries. The effect on separation bubbles and vortices moving through these regions must be examined. It should be noted that all of these concerns are typical for any flow solver using domain decomposition. Finally, it is noted that the viscous terms in this code are not parallelized. However, it will not impact our approach, as we will be replacing them with vorticity confinement terms.

Blade Motion Modifications to TURNS_SERIAL

A two-degree-of-freedom motion capability (i.e. pitch and plunge) was added to TURNS_SERIAL through blade motion terms. Adding this feature involved computing the appropriate time-dependent metrics terms for the associated motion. These metrics occur as terms in a general curvilinear formulation of the equations of motion and are defined as follows:

$$\begin{bmatrix} \xi_t \\ \eta_t \\ \zeta_t \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \qquad \textbf{(2)}$$

The grid point velocities are

$$\{\dot{X}\}^{n+1} = \frac{1}{\Delta t}\left(\{X\}^{n+1} - \{X\}^{n}\right) \qquad \textbf{(3)}$$

and $\{X\} = \lfloor x \quad y \quad z \rfloor^T$ are grid point locations. After each time step of the flow solver, (1) a new grid position is calculated, either from a prescribed motion or as computed from the solution of some structural dynamics equations, and the grid is moved to its new position, (2) grid velocities are calculated for each point, (3) new spatial metric terms are calculated, and (4) new time metric terms are calculated. The flow solver then proceeds through the computation of the next time step.

Vorticity Confinement

Vorticity Confinement is a unique method that has been developed by FAI over the last decade. It was initially conceived out of a motivation to mitigate the effects of the inherent dissipation in numerical schemes for the Euler and Navier-Stokes equations. These effects result in a rapid diffusion of concentrated regions of vorticity. When these schemes are applied to rotorcraft problems, which routinely operate in their own wake, there is a definite loss of accuracy associated with the inability to preserve these structures over the required distances. Basically, the method was used as a way to add an anti-diffusive term to the equations without affecting stability. Vorticity confinement was effectively incorporated into a solver as a source term in the momentum equations. Early applications of the method quickly demonstrated a dramatic improvement in the lengths to which vortical structures could be maintained.

In the last few years, there have been a number of improvements in the technique for implementation, as well as an improved understanding of the more general theory it involves and to what it may be applied **[**22**].** To succinctly state it, the technique can be considered as a new and very different type of model for concentrated thin vortical regions that is far more efficient than PDE-based methods. This is the foundation of the present paper because, without it, first-principles-based methods must resolve these regions with an enormous number of grid points and associated computational expense.

Incompressible Flow Implementation

In the implementation used in the incompressible solvers developed by FAI, the confinement term is introduced as a force in the momentum equations and, as part of a multiple step procedure, is actually added as a velocity correction at one of the steps. For general unsteady incompressible flows, the governing equations with the diffusive and corrective vorticity confinement terms that accompany it are:

$$\partial_t \vec{q} = -(\vec{q}\cdot\nabla)\vec{q} + \nabla(p/\rho) + \mu\nabla^2\vec{q} - \varepsilon\vec{s} \qquad \textbf{(4)}$$

$$\nabla\cdot\vec{q} = 0 \qquad \textbf{(5)}$$

Here, $\vec{q}$ is the velocity vector, p is pressure, ρ is density, and μ is a numerical diffusion coefficient. The diffusion term does not represent physical diffusion, but is implemented to provide isotropic diffusion and balance the confinement term. The second confinement term is anti-diffusive, or corrective, and ε is a numerical coefficient, which together with μ, controls the size of the convecting vortical regions or vortical boundary layers. There are many possible forms for the confinement term. The simplest one is given by

$$\vec{s} = \vec{n}\times\vec{\omega} \qquad \textbf{(6)}$$

$$\vec{n} = \vec{\nabla}\eta \,/\left|\vec{\nabla}\eta\right| \qquad \textbf{(7)}$$

$$\vec{\omega} = \vec{\nabla}\times\vec{q} \qquad \textbf{(8)}$$

where $\vec{\omega}$ is the vorticity vector, and the scalar field, η ,is defined as $\eta = \left|\vec{\omega}\right|$.

For the corrective term, $\vec{n}$ is a unit vector pointing into the centroid of the vortical region and the term serves to convect vorticity back towards the centroid as it diffuses away. This convection increases the diffusion term and a steady-state form results when the two terms become balanced. It is noted that steady-state solutions exist, for any positive value of ε. For the present types of flows, it appears to be better to discretize Equations (4-8) for problems that have a thin, well-behaved vorticity distribution, even in the presence of the numerical diffusion, than to discretize the unmodified equations, which only admit vorticity regions that continue to spread, if there is any numerical diffusion.

An important feature of the vorticity confinement method is that the extra terms are generally limited to the vortical regions. Another important feature concerns the total change induced by the correction in mass, vorticity, and momentum, when integrated over a cross section of a convecting vortex. It is shown in Ref. **[**23**]** that mass and vorticity are explicitly conserved and momentum is almost exactly conserved. (A small extension of the method allows an exact preservation).

In general, computed flows do not depend sensitively on the parameters, ε and μ, for a range of values. Hence, the issues involved in setting them are similar to those involved in setting numerical parameters in other standard computational fluid dynamics schemes, such as artificial dissipation in many conventional compressible solvers. The reason for this lack of sensitivity is that, if a vortex is axisymmetric, the velocity outside the core is not sensitive to the vorticity distribution, as long as the radius is kept small and prevented from becoming large due to numerical effects. It has been shown numerically that vortical solutions to the discretized equations are close to those predicted for the continuum ones, even though the vortical regions are only a few cells thick. Roughly speaking, the confinement term seems to be convecting discretization errors into the vortex center. This point should be addressed by an analysis of the discrete equations themselves, which is currently under investigation. Finally, it should be mentioned that these solutions should be considered as *zeroeth-order* solutions, which are very economical but do not take into account dynamics in the vortical cores, such as turbulence.

Boundary Layer Modeling

When this method is applied to viscous surface flow computations, like dynamic stall, vorticity confinement can be thought of as a crude boundary layer model. Every time-step, an acceleration is applied to the boundary layer along vorticity contour lines (i.e., in the tangential direction). Together with numerical diffusion, this results in a vortical region spread over 2-3 grid cells as it convects over the surface. This artificial boundary layer acts like a turbulent boundary layer. The confined computational layer can convect into a region of adverse pressure gradient some distance before separating. One effect of real, physical turbulence is to accelerate the upper part of the viscous sub-layer in the tangential direction. This is similar to the action of numerical confinement. Accordingly, the turbulent layer does not undergo a laminar-type low Reynolds number separation. By contrast, a laminar, finite thickness boundary layer would thicken in an adverse pressure gradient. This thickening would lead to acceleration of the flow over the thickening spot and lower pressure, which would further increase the thickening, resulting in rapid separation.

The physical turbulent sublayer, like the numerical vorticity confinement layer, tends to maintain an unchanged thickness until a separation occurs from within the layer. This separation can have a very simple cause - the *collapse* of fluid elements in the longitudinal direction because of their viscosity and pressure induced momentum loss. This results, through incompressibility, in an *eruption* in the normal direction **[**24**]**. This description is, of course, heuristic. A more precise analysis is currently under development, which should lead to rational models for separation, including finite Reynolds number effects.

Compressible Flow Implementation

The first implementation of vorticity confinement in a compressible solver (TURNS) **[**25**]** employed a velocity correction approach. The most generalized compressible flow implementation has been developed in the two-dimensional Euler/Navier-Stokes code ARC2D **[**26**]**. The vorticity confinement terms in this code have been incorporated either as a body force or a velocity correction term and are considered either within or external to the sub-iteration process. However, either approach is satisfactory and they can be shown to be equivalent to first order, provided an energy correction is made for compressible flows. The implementation in TURNS_SERIAL can then be presented as a modification to the governing equations (Eqn. 1) discussed above where the viscous term, $\boldsymbol{G_V}$, is dropped (Euler assumption), and the body force term, $\boldsymbol{B}$, is defined as

$$\boldsymbol{B} = \delta\boldsymbol{Q}_d + \delta\boldsymbol{Q}_c \qquad \textbf{(9)}$$

The right-hand-side of Eqn. 9 consists of the diffusive and corrective terms of vorticity confinement. The details of the implementation are described in **[**25**]**.

Two-Degree-of-Freedom Structural Dynamics Model

The two-degree-of-freedom structural dynamics model for an airfoil is classically described by the typical section in aeroelastic literature **[27].**

The discussion here follows **[28]** very closely. The parameters that define this model are detailed there and shown in Figure 1. The plunging motion is represented by the displacement $\boldsymbol{h}$ (positive downward), and pitching motion by α (positive nose-up and measured about the elastic axis). The elastic axis is measured rearward from the mid-chord by the distance $\boldsymbol{a_h b}$ and the mass center is at $\boldsymbol{x}_\alpha$ measured rearward from the elastic axis. Assuming a rigid airfoil, small amplitude oscillations, and no coupling in the mechanical damping terms, the equations of motion of the system are written as:

$$S_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + C_\alpha \dot{\alpha} + K_\alpha \alpha = Q_\alpha \qquad \textbf{(10a)}$$

$$m\ddot{h} + S_\alpha \ddot{\alpha} + C_{\boldsymbol{h}} \dot{h} + K_h h = Q_h \qquad \textbf{(10b)}$$

where $\boldsymbol{m}$ is the airfoil mass, $\boldsymbol{I}_\alpha$ is the polar moment of inertia about the elastic axis, $\boldsymbol{S}_\alpha$ is the static mass moment about the elastic axis, $\boldsymbol{C}_\alpha$ and $\boldsymbol{C_h}$ are the mechanical damping coefficients for pitch and plunge, respectively. $\boldsymbol{K}_\alpha$ and $\boldsymbol{K_h}$ are the spring constants corresponding to pitch and plunge, respectively, and the loads are given by

$$\begin{Bmatrix} Q_h \\ Q_\alpha \end{Bmatrix} = \tfrac{1}{2}\rho_\infty U_\infty^2 c \begin{Bmatrix} C_L \\ cC_M \end{Bmatrix} \qquad \textbf{(11)}$$

The mathematical formulation for the structures model is classical, assuming a rigid airfoil and small amplitude oscillations. If the airfoil is constrained to move only in two dimensions (i.e., allowed to plunge and pitch only), the equations of motion may be written:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F\} \qquad \textbf{(12)}$$

where $u = \lfloor \bar{h} \quad \alpha \rfloor^T$, $\bar{h} = h/b$ and α are the vertical plunge translation and pitch angle, respectively. **[*M*], [*C*],** and **[*K*]** are the mass, damping, and stiffness matrices, respectively, and $\boldsymbol{F}$ is the forcing function given by

$$\{\boldsymbol{F}\} = \frac{4}{\pi \mu_S \boldsymbol{k_c^2}} \begin{Bmatrix} -C_L \\ 2C_M \end{Bmatrix} \qquad \textbf{(13)}$$

where $\boldsymbol{C_L}$ and $\boldsymbol{C_M}$ are the lift and moment coefficients, respectively, calculated by the aerodynamics module, and $\boldsymbol{k_c}$ is the reduced frequency based on half chord. The variable μ_s is a constant that relates the airfoil mass to the air mass, and is given by

$$\mu_s = \frac{I_\alpha}{\pi \rho b^4} \qquad \textbf{(14)}$$

These equations are solved in the time domain using finite difference schemes. For the present work, the damping coefficients are considered negligible. An implicit and an explicit scheme are simply derived and given as

Implicit

$$\{u\}^{n+1} = \left([I] + \Delta t^2 [M]^{-1}[K]\right)^{-1} \left(2\{u\}^n - \{u\}^{n-1} + \Delta t^2 [M]^{-1}\{F\}^n\right)$$

(15a)

Explicit

$$\{u\}^{n+1} = 2\{u\}^n - \{u\}^{n-1} + \Delta t^2 [M]^{-1}\left(\{F\}^n - [K]\{u\}^n\right)$$

(15b)

The coupled solution procedure is described as follows. A computational step is performed by the flow solver, after which the forces and moments are computed. The loads are sent to the structural equations solver and the new displacements computed. These displacements are then used to compute a new airfoil position and, thus grid, and the process repeats until the desired number of cycles of motion are completed.

Computed Results

Dynamic Stall Computations

For validation of the method, several test cases were chosen to exercise the vorticity confinement method, as implemented in TURNS_SERIAL, over a range of pitch motions corresponding to fully attached flow through deep dynamic stall. With conventional CFD approaches, computations of this type would require the resolution of the boundary layer, in addition to the use of a turbulence model. The dense grid resolution required near the airfoil results in enormous computational resource requirements that render the use of these methods infeasible, particularly if coupled to a comprehensive code in a production environment. With vorticity confinement, however, calculations of this type can be performed on coarse grids, with grid refinement generally no greater than that required for Euler inviscid calculations

The grid used in the following studies is depicted in Fig. 2. The mesh is a C-grid for a NACA 0015 airfoil, consisting of 141 points around the airfoil, and 41 points extending normally to the far field, for a total of approximately 6500 points. The coarse nature of the grid can clearly be seen in the figure; the grid resolution near the surface is orders-of-magnitude coarser than what would be required for conventional viscous computations. No-slip boundary conditions are used at the body surface which, in addition to the coarse grid, generates a massive vorticity field and nonphysical separation, in the absence of confinement. However, the physical properties of the boundary layer can be modeled through the judicious choice of the diffusion and confinement parameters, μ and ε, respectively.

Attached Flow

The computations were performed for a free-stream Mach number of 0.29. Three main flow conditions were addressed: attached flow, light stall, and deep stall. The Euler equations (with the vorticity confinement corrections) were solved; no viscous terms or conventional turbulence models were employed. Vorticity is generated at the surface by applying no-slip (zero velocity) conditions. Each pitching cycle used 1440 flow solver time steps, with three sub-iterations at each time step. The computed results are compared against data the data of Piziali [29].

Pitching the airfoil between angles of attack of -0.2 and 8.2 degrees results a flow field that remains fully attached. Results of the attached flow computations, presented as lift and moment hysteresis curves, are shown in Figs. 3 and 4. In this benign case, the vorticity confinement results lie very close to the experimental data.

Light Stall

The light stall computation is of particular interest because in many ways it is more difficult to compute than deep stall. Flows exhibiting light stall are inherently metastable, and very small fluctuations in flow conditions can radically alter the resulting flow field. Figs. 5 and 6 depict lift and moment hysteresis curves for a light stall case, with angle of attack varying between 6.8 to 15.2 degrees. The agreement is reasonable, but was found to be more sensitive to values of the confinement parameter than both the attached flow and deep stall cases.

Deep Stall

Finally, a deep stall case is presented representing results when the airfoil is pitched between 10.8 and 19.2 degrees. The resulting hysteresis curves for lift and moment are presented in Figs. 7 and 8. As in the previous results, the agreement between the computations and experiment is very good, especially considering the coarse grid, the relative simplicity of the approach, and the large time steps taken.

T demonstrate the benefits of the present method, two cases were run *without* vorticity confinement. Fig. 9 depicts lift hysteresis curves for two Euler solutions; one using standard slip boundary conditions, and one using no-slip conditions. The latter solution represents simply setting the confinement parameters of the deep stall case to zero. Both unconfined solutions are grossly in error compared to the experimental results. The Euler solution fails to stall (except for evidence of a light stall, due to the inherent dissipation in the solution algorithm). The no-slip condition results in massive flow separation that never reattaches. Comparison of the results of Fig. 9 with Fig. 7

dramatically illustrates the ability of vorticity confinement to model the deep stall case.

The test cases demonstrate the capability of the vorticity confinement approach for the computation of flows of interest in the current paper (albeit in 2-D). It must be stressed that these computations are Euler calculations done on coarse grids, and it is the vorticity confinement approach that allows flows of this nature to be computed. The results are in all cases at least as good, and in some cases better, than similar stall computations done using Navier-Stokes calculations with turbulence modeling. In addition, conventional calculations are found to require roughly two orders of magnitude greater computational time than the vorticity confinement approach. The stall cases presented herein required roughly 2.5 minutes per cycle to compute on a Cray C90 (the same computations were performed on a Pentium III/700Mhz PC in approximately 10 minutes per cycle).

Coupled Aerodynamic and Structural Dynamics Computations

The typical coupled solution procedure requires force and moment information to be sent to the structures module as forcing functions, while displacement information is sent to the aerodynamics module for appropriate repositioning of the grid. The cycle then repeats for each time step. The coupled computations discussed below were performed using TURNS_SERIAL for a NACA 0015 airfoil oscillating in pitch only.

Comparison of Structures Solvers

As described in above, both an implicit and an explicit difference scheme were coded for the integration of the structural dynamics equations. To isolate the numerical properties of the schemes, the computations were performed in free-oscillation using only the structures solution methods. The pitch angle was initialized at 15^o and each of the schemes was used to compute the angle. With no structural or aerodynamic damping the solution expected would be a perfect sinusoid. Time steps corresponding to 0.25^o and 5.0^o for the aerodynamics code were used.

The results for the implicit scheme are shown in Fig. 10. Even for the small time step, which typical of aerodynamics solvers, there is approximately a 1.5% dissipation of the peak amplitude per cycle of oscillation. However, for the large time step case, which may be typical of structures solvers, the dissipation rate is about 25% per cycle. Figure 11 depicts the merits of the explicit scheme. Here there is virtually no difference between the small and large time step cases and the dissipation rate is approximately 0.01%. Clearly, the explicit scheme is far superior and was used for all subsequent computations.

Aerodynamically Damped Solutions

As a result of the low dissipation of the structures integration scheme, any damping observed in the following computations is guaranteed to be a result of aerodynamic or structural effects only. Fully coupled solutions between the TURNS_SERIAL code and the simple 2 DoF structural model were computed using the same time step for the aerodynamics and structures code. A Mach number of 0.3 was used and the angle of attack was initiated at 5^o.

A number of computations were performed with different values of the aero-structures parameter μ_s, defined in Eqn. 14. The value of this parameter determines the amount of aerodynamic damping [27]. Large values of this parameter correspond to a solid steel airfoil oscillating in thin air and thus no damping would be expected.

Figure 12 shows the effect μ_s on the solutions for the pitch angle for two cycles of oscillation. They range from undamped to highly damped. Typical values of this parameter are around 50. The coupled procedure demonstrates the proper variation of the pitch angle with μ_s and gives confidence in the procedure.

Modified Time-stepping Procedure

Typical time steps required by aerodynamics (CFD) codes are substantially less (0.25^o of azimuth for rotor blade solutions) than that for structural dynamics codes (5^o). For an effective coupling procedure, the ability to use different time stepping is necessary. To demonstrate this capability, solutions for two different time step ratios were used. The first ratio, $\Delta t_s/\Delta t_a$ was 1 and corresponded to that used for the coupled solutions above. The second ratio was 20 and

represents the desired ratio for coupling of a CFD code to a comprehensive rotor code. For the case where the time steps are different, the pitch angle was computed by a linear interpolation between the previous angles predicted by the structures codes to allow for a smooth grid motion.

The solutions were computed using the same initial conditions and Mach number as in the previous section and the value of μ_s was 50. The results are shown in Fig. 13 for 5 cycles of oscillation. There is very little difference between the two solutions. However there are some phase and amplitude differences. They are most noticeable during the first few cycles and are considerably reduced by the last few cycles. This may be due to a much different procedure used for setting the initial conditions for the two methods for predicting the pitch angle. Nonetheless, the approach is most promising for future research into coupling CFD and comprehensive codes for rotorcraft predictions.

Scalability Calculations

As scalability issues are inapplicable to the serial code used above, the parallel code Turns_Mpi was installed and executed on an Origin 2000 system that employs a distributed memory configuration. Tests were performed on both two- and three-dimensional geometries to ascertain its scalability as well as applicability to the current and future efforts. Effective use of Turns_Mpi in a comprehensive environment will require good scalability up to approximately 50 processors. This has been demonstrated for three-dimensional computations using Turns_Mpi on another machine [21] for a much larger number of processors.

Airfoil Computations

The steady-state flow field over a NACA 0015 airfoil was computed using a grid consisting of approximately 7700 points (167x46). Computations were performed using 1, 5, 7, and 11 processors. The results of the scalability study are depicted in Fig. 14. Near-linear speedup is observed for the 3-processor case; however, less dramatic speedups are evident for larger numbers of processors. Note that this study employed a much smaller grid that the earlier study mentioned above. As a result, the boundary/volume grid point ratio was much smaller and thus the scalability is somewhat diminished.

Wing Computations

Steady-state computations were also performed for a default case, an OLS wing using a grid (135 x 50 x 35) of approximately 240,000 points. The number of processors used in the study varied from 1 to 21 and the results are shown in Fig. 15. Excellent scalability is shown over the entire range of processors used. This should be a very adequate candidate for future coupling with comprehensive codes in a parallel environment.

Conclusions

A general two-degree-of-freedom motion capability and the vorticity confinement method have been added to the three-dimensional Euler/Navier Stokes rotor code, Turns_Serial. The resultant code has been used to predict various stall events for a NACA 0015 airfoil with excellent comparisons to experimental data. Results were obtained in one to two orders-of-magnitude less CPU time than current Navier-Stokes modeling, demonstrating the efficiency of the method. A simple two-degree-of-freedom structural dynamics model has been tightly coupled to this code and aerodynamically damped, stable solutions obtained for a range of problem parameters (pitch motion only). Scalability studies have been performed using a parallel construction of the flow solver, Turns_Mpi, for two- and three-dimensional steady flows (using a fixed problem size). Scalability was shown for approximately 5 to 7 processors for the two-dimensional case. Far more relevant to the long-term goals of the work is the fact that excellent scalability was shown past 20 processors for the three-dimensional application.

The results achieved in this work have demonstrated that: a) a suitable compressible flow solver combined with an efficient dynamic stall model, provided through the use of vorticity confinement, has been developed; b) tight integration of the aerodynamics and structures

equations has been demonstrated to be stable, even with vastly different time-steps for the two disciplines; and c) the parallel version of the flow solver demonstrates excellent scalability, particularly for three-dimensional applications.

The work completed to date has provided confidence that there is a high probability for success for our future work, which involves coupling an established comprehensive code with the parallel TURNS_MPI Euler/Navier Stokes code, with vorticity confinement modifications for stall modeling. The flow solver will be used as an isolated blade code, with inflow, prescribed motion, and blade deformation modeled through gridding changes, and with no attempt to resolve the wake (in fact, disabled through the use of wing type boundary conditions to avoid double definition), and the use of the lift and moment data in a tight coupling. Applications to existing data, such as that available from the UH-60A Airloads Program, should provide sufficient validation for the approach.

References

[1] Maier, T., and Bousman, W., "An Examination of the Aerodynamic Moment on Rotor Blade Tips Using Flight Test Data and Analysis," *18th European Rotorcraft Forum*, Sept. 1992.

[2] Coleman, C.P., and Bousman, W.G., "Aerodynamic Limitations of the UH-60A Rotor," NASA TM-110396, USAATCOM Technical Report 96-A-011, August 1996.

[3] Kufeld, R.M., and Johnson, W., "The Effects of Control System Stiffness Models on the Dynamic Stall Behavior of a Helicopter," presented at the *American Helicopter Society 54th Annual Forum*, Washington, DC, May 20-22, 1998.

[4] Chandrasekhara, M.S., and Carr, L.W., "Compressibility Effects on Dynamic Stall of Oscillating Airfoils," AGARD-CP-552, pp. 3:1-3:15, August 1995.

[5] McCroskey, W.J., McAlister, K.W., Carr, L.W., Pucci, S.L., Lambert, O., and Indegrand, "Dynamic Stall on Advanced Airfoil Sections," *AHS Journal*, 26,3, July 1981, pp. 40-50.

[6] Bousman, W.G., "A Qualitative Examination of Dynamic Stall from Flight Test Data," *Journal of the American Helicopter Society*, Vol. 43, No. 4, pp. 279-295, October 1998.

[7] Johnson, W., "Rotorcraft Aerodynamics Models for a Comprehensive Analysis," presented at the *American Helicopter Society 54th Annual Forum*, Washington, DC, May 20-22, 1998.

[8] Nguyen, K., and Johnson, W., "Evaluation of Dynamic Stall Models with UH-60A Airloads Flight Test Data," presented at the *American Helicopter Society 54th Annual Forum*, Washington, DC, May 20-22, 1998.

[9] Ko, S. and McCroskey, W.J., "Computations of Unsteady Separating Flows Over an Oscillating Airfoil," AIAA Paper No. 95-0312, presented at the *33rd Aerospace Sciences Meeting and Exhibit,* Jan.9-12, 1995, Reno, NV.

[10] Bangalore,A.K., Moulton,M.A., and Caradonna,F.X., "The Development of an Overset/Hybrid Method for Rotorcraft Applications," presented at the *American Helicopter Society Technical Specialists Meeting for Aerodynamics and Aeroacoustics*, Williamsburg, VA, Oct. 27-30, 1997.

[11] Moulton, M., and Steinhoff, J.S. "A Technique for the Simulation of Stall Coarse-Grid CFD Methods," AIAA-2000-0277, *AIAA Aerospace Sciences Meeting*, Reno, NV, January, 2000.

[12] Tung, C., Caradonna, F.X., and Johnson, W., "The Prediction Transonic Flows on Advancing Rotors," *Journal of the American Helicopter Society*, Vol.31, July 1986, pp4-9.

[13] Strawn, R.C., and Caradonna, F.X., "Conservative Full Potential Model for Unsteady Transonic Rotor Flows," *AIAA Journal*, Vol. 25, No. 2, p193, February 1987.

[14] Bridgeman, J.O., Steger, J.L., and Caradonna, F.X., "A Conservative Finite-Difference Algorithm for the Unsteady Transonic Potential Equation in Generalized Coordinates," AIAA Paper No. 82-1388, presented at the *AIAA 9th Atmospheric Flight Mechanics Conference*, San Diego, California, Aug. 1982.

[15] Strawn, R.C., Desopper, A., Miller, Judith, Jones, Alan, "Correlation of Puma Airloads Evaluation of CFD Prediction Methods," Paper No.14, *Fifteenth European Rotorcraft Forum*," Sept 1989.

[16] Strawn, R.C., and Bridgeman, J.O., "An Improved Three-Dimensional Aerodynamics Model for Helicopter Airloads Prediction," presented at the *AIAA 29th Aerospace Sciences Meeting*, Reno, Nevada, Jan. 7-10, 1991.

[17] Lee, C.S., Saberi, H., and Ormiston, R.A., "Aerodynamic and Numerical Issues for Coupling CFD into Comprehensive Rotorcraft Analysis," presented at the *American Helicopter Society 53rd Annual Forum*, Virginia Beach, VA, April 29-May 1, 1997.

[18] Bridgeman, J.O., Prichard, D., and Caradonna, F.X., "The Development of a CFD Potential Method for the Analysis of Tilt-Rotors," presented at the *AHS Technical Specialists Meeting on Rotorcraft Acoustics and Fluid Dynamics*, Philadelphia, PA, October 15-17, 1991.

[19] Lim, J.W., Panda, B., Sopher, R., Cassarino, S.J., and Lee, C.S., "Current Status of U.S. Army Comprehensive Helicopter Analysis System," presented at the *American Helicopter Society 56th Annual Forum*, Virginia Beach, VA, May 2-4, 2000.

[20] Srinivasan, G. R., Baeder, J. D., Obayashi, S. and McCroskey, W. J., "Flowfield of a Lifting Rotor in Hover: A Navier-Stokes Simulation," *AIAA Journal*, 30, No.10 (Oct.1992): 2371-2378.

[21] Wissink, A.M., Lyrintzis, A.S., Strawn, R.C., Oliker, L., and Biswas, R., "Efficient Helicopter Aerodynamic and Aeroacoustic Predictions on Parallel Computers," AIAA Paper No. 96-0153, presented at the *AIAA 34th Aerospace Sciences Meeting*, January 15-18, 1996, Reno, Nevada.

[22] Steinhoff, J., Puskas, E., Babu, S., Wenren, Y., and Underhill, D., "Computation of Thin Features Over Long Distances Using Solitary Waves," *AIAA Proceedings*, 13th Computational Fluid Dynamics Conference, July 1997.

[23] Steinhoff, J., Wang, C, Underhill, D., Mersch, T., and Wenren, Y., "Computational Vorticity Confinement: A Non-diffusive Eulerian Method for Vortex-dominated Flows," UTSI Preprint, 1992.

[24] Van Dommelen, L.L. and Shen, S.F., "The Spontaneous Generation of the Singularity in a Separating Laminar Boundary Layer," *Journal of Computational Physics*, 38:12, 1980.

[25] Wang, C.M., Bridgeman,J.O., Steinhoff,J.S. and Wenren,Y., "The Application of Computational Vorticity Confinement to Helicopter Rotor and Body Flows," presented at the *American Helicopter Society 49th Annual Forum*, St. Louis, MO, May19-21, 1993.

[26] Pulliam, T. H., "Efficient Solution Methods for the Navier--Stokes Equations." Lecture Notes for the Von Karman Institute for Fluid Dynamics Lecture Series: Numerical Techniques for Viscous Flow Computation in Turbomachinery Bladings, January 20-1986, Brussels, Belgium.

[27] Bisplinghoff, R., Ashley, H, and Halfman, R., Aeroelasticity, Addison-Wesley, 1955.

[28] Yang, T.Y., Guruswamy, P., and Striz, A.G., "Flutter Analysis of a NACA 64A006 Airfoil in Small Disturbance Transonic Flow," *Journal of Aircraft* , Vol. 17, No. 4, pp. 225-232, April 1980.

[29] Piziali, R.A., "2-D and 3-D Oscillating Wing Aerodynamics for a Range of Angles of Attack Including Stall". Technical Report, NASA TM-4632, 1994.

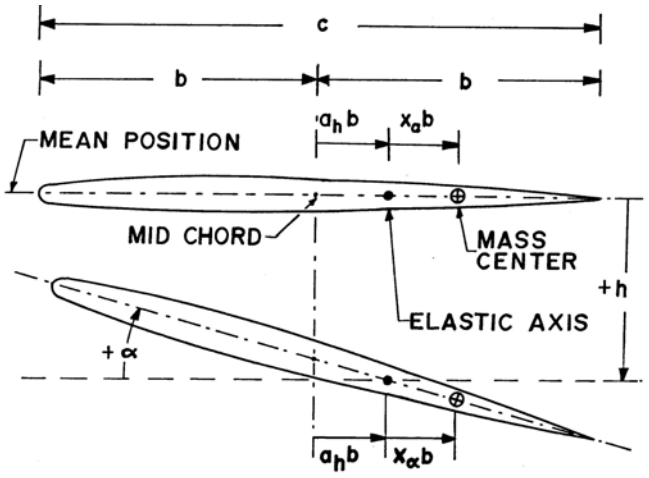

Figure 1. Typical section structural dynamics model (two-degrees-of-freedom) .

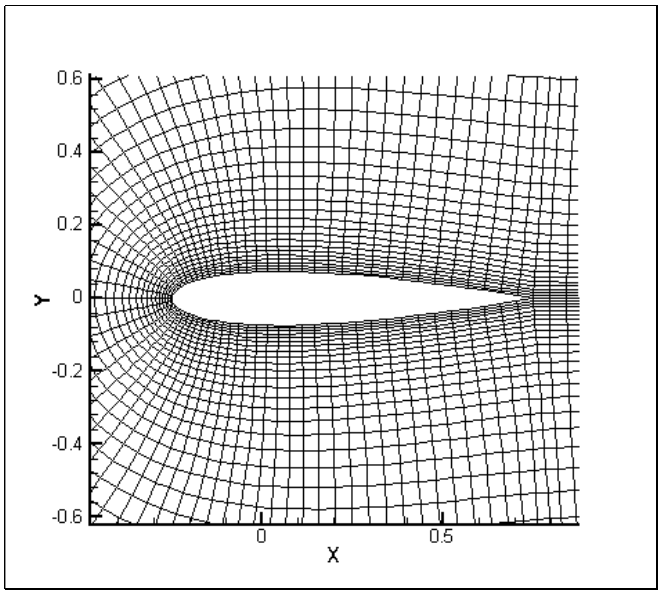

Figure 2. NACA 0015 Airfoil C-Grid Used for Dynamic Stall Computations (141x46).

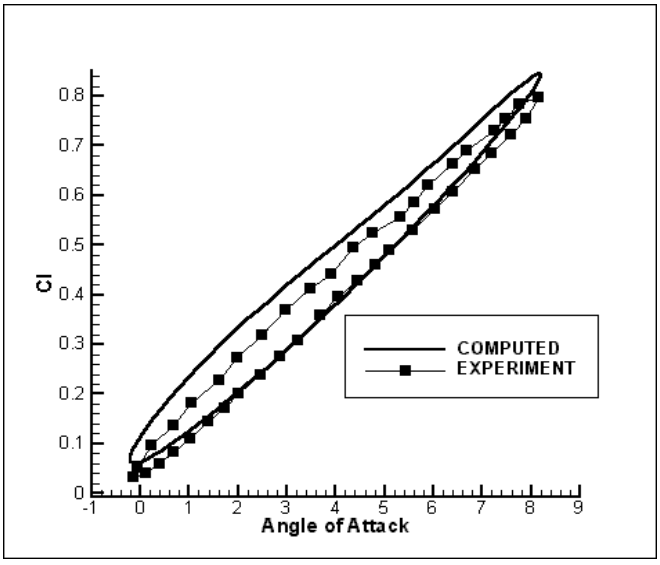

Figure 3. Lift Hysteresis Prediction for the NACA 0015 Airfoil in Attached Flow.

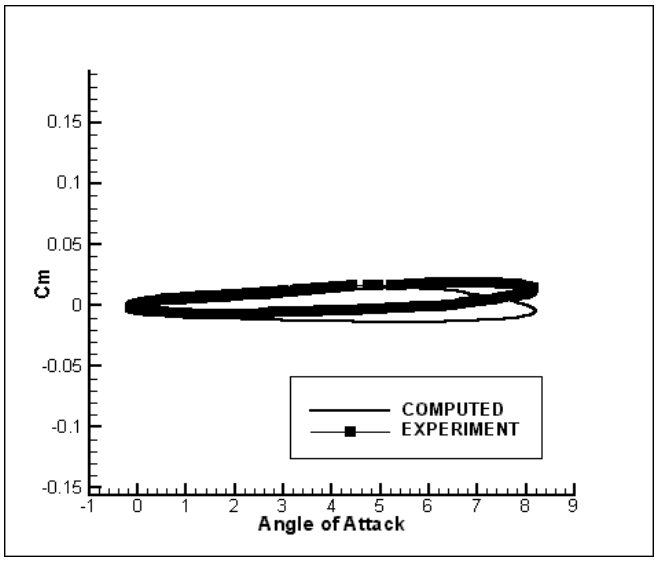

Figure 4. Moment Hysteresis Prediction for the NACA 0015 Airfoil in Attached Flow.

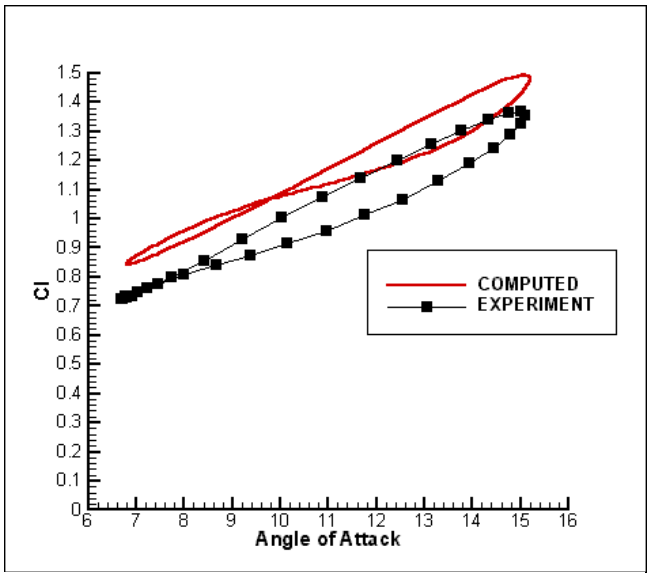

Figure 5. Lift Hysteresis Prediction for the NACA 0015 Airfoil in Light Stall.

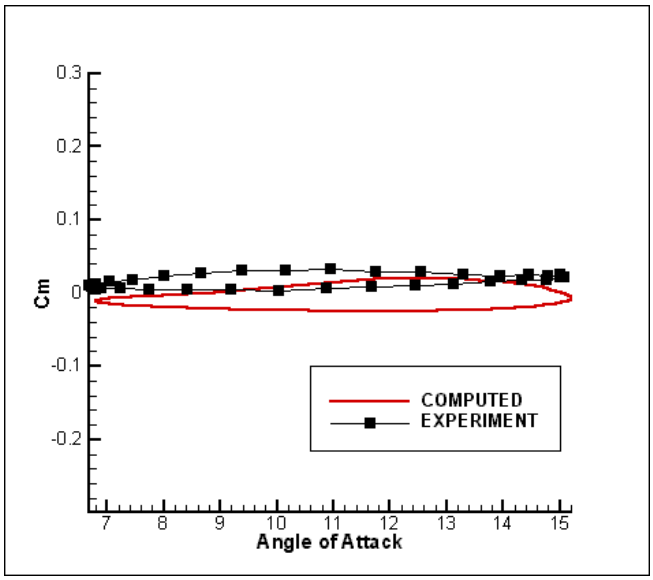

Figure 6. Moment Hysteresis Prediction for the NACA 0015 Airfoil in Light Stall.

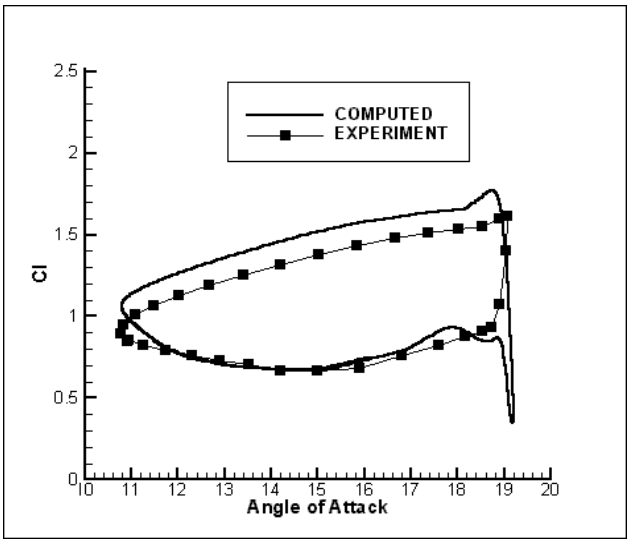

Figure 7. Lift Hysteresis Prediction for the NACA 0015 Airfoil in Deep Stall.

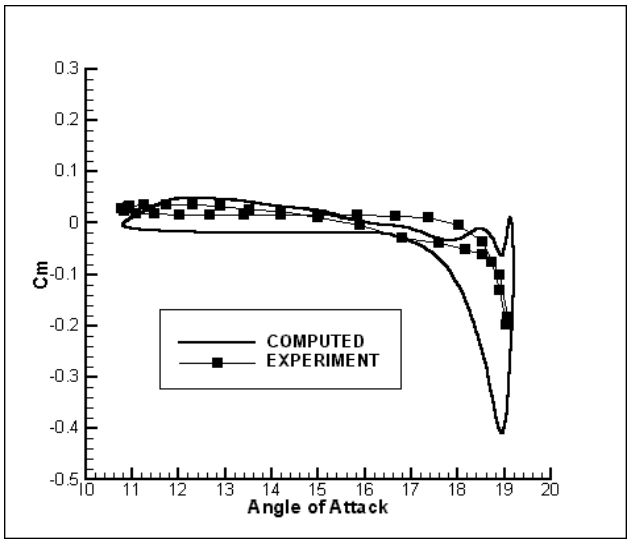

Figure 8. Moment Hysteresis Prediction for the NACA 0015 Airfoil in Deep Stall.

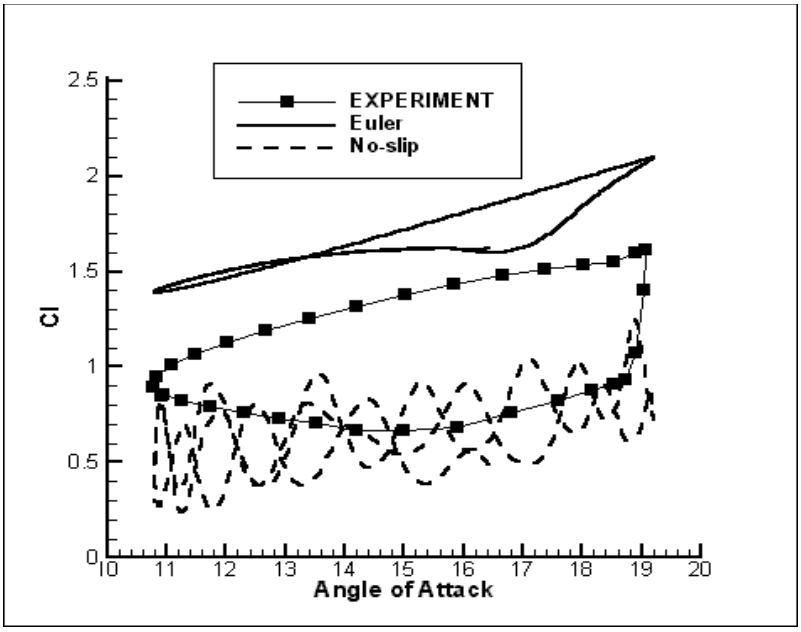

Figure 9. Predicted Lift Using Turns_Mpi Without Vorticity Confinement using Slip (Euler) and No Slip Body Surface Boundary Conditions.

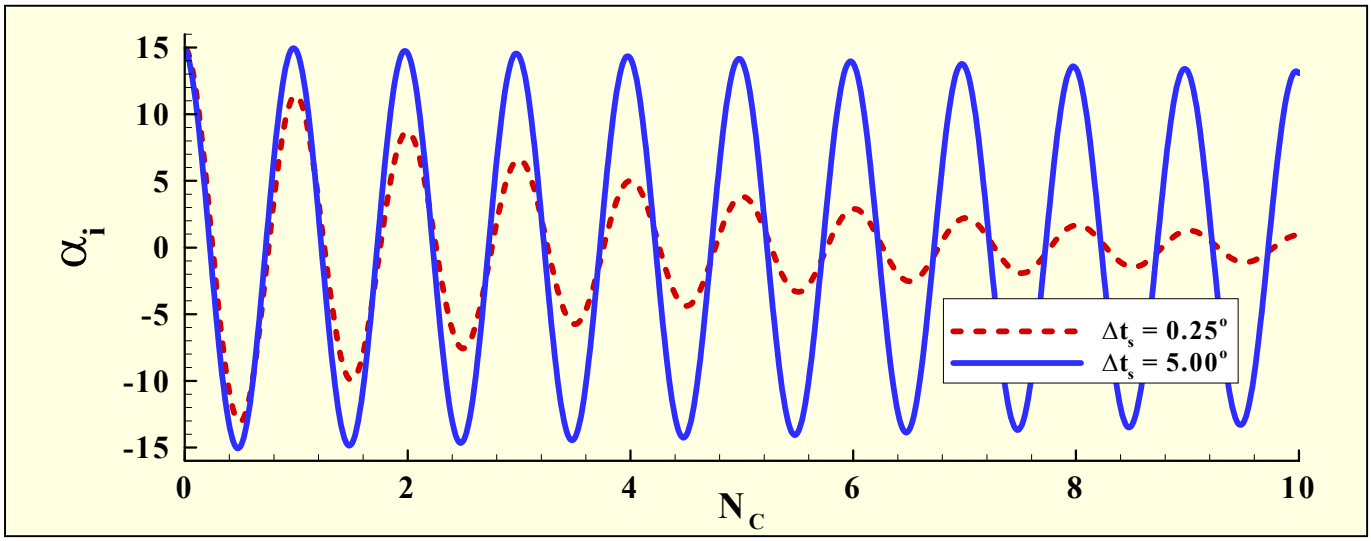

Figure 10. Time History of the Pitch Deflection Free-oscillation of a NACA 0015 Airfoil Using an Implicit Method for the Structural Dynamics Solver.

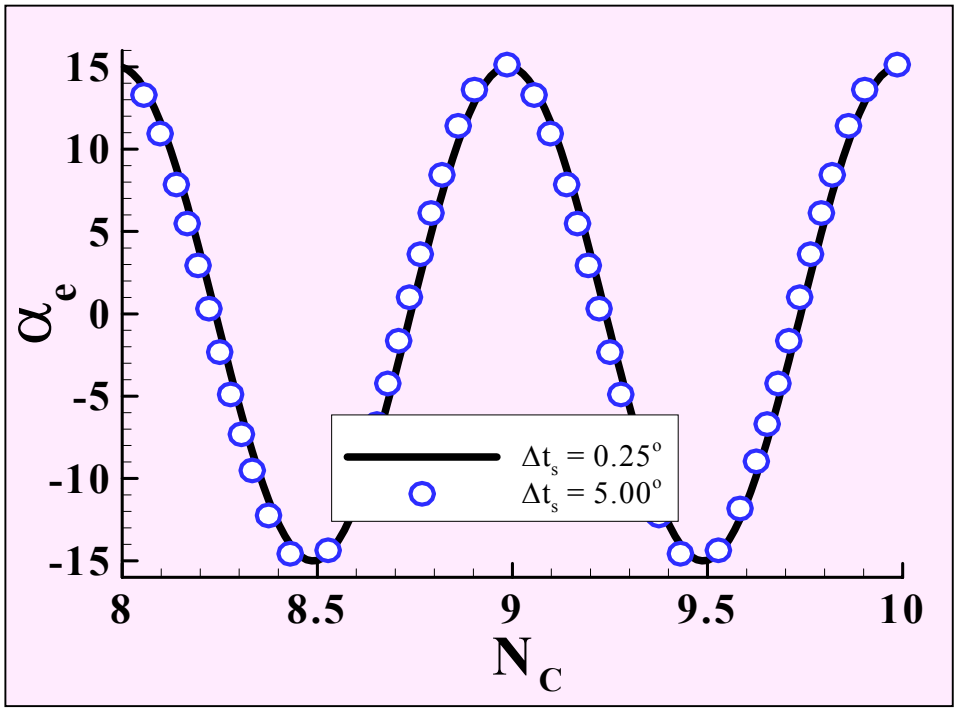

Figure 11. Time History of the Pitch Deflection Free-oscillation of a NACA 0015 Airfoil Using an Explicit Method for the Structural Dynamics Solver.

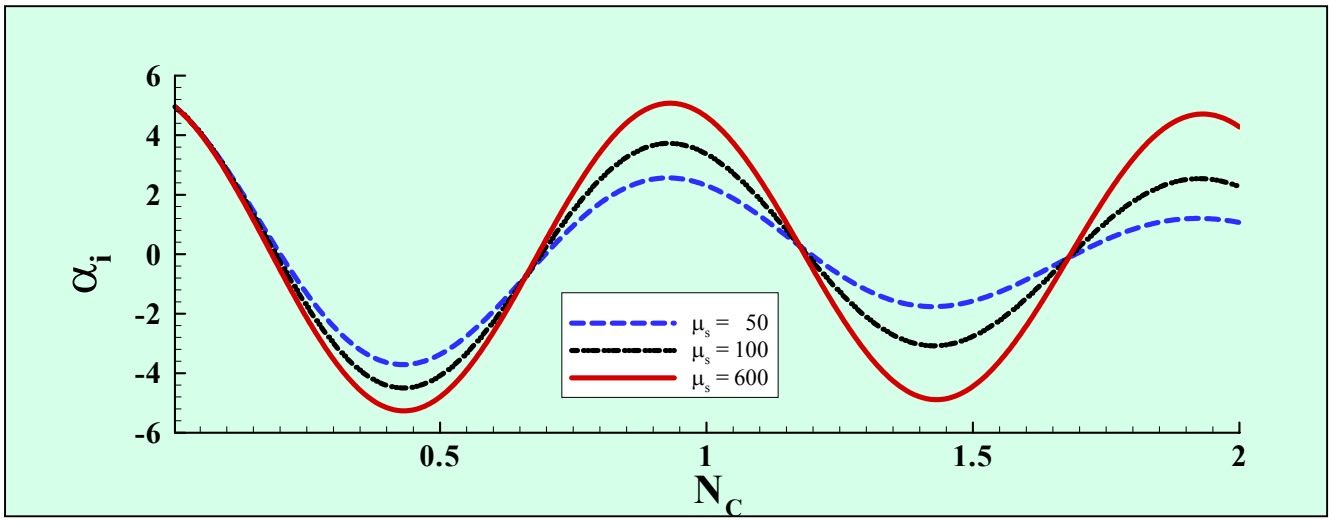

Figure 12. Variation of Aerodynamic Damping with μ_s for Coupled Aerodynamic and Structural Dynamics Solutions.

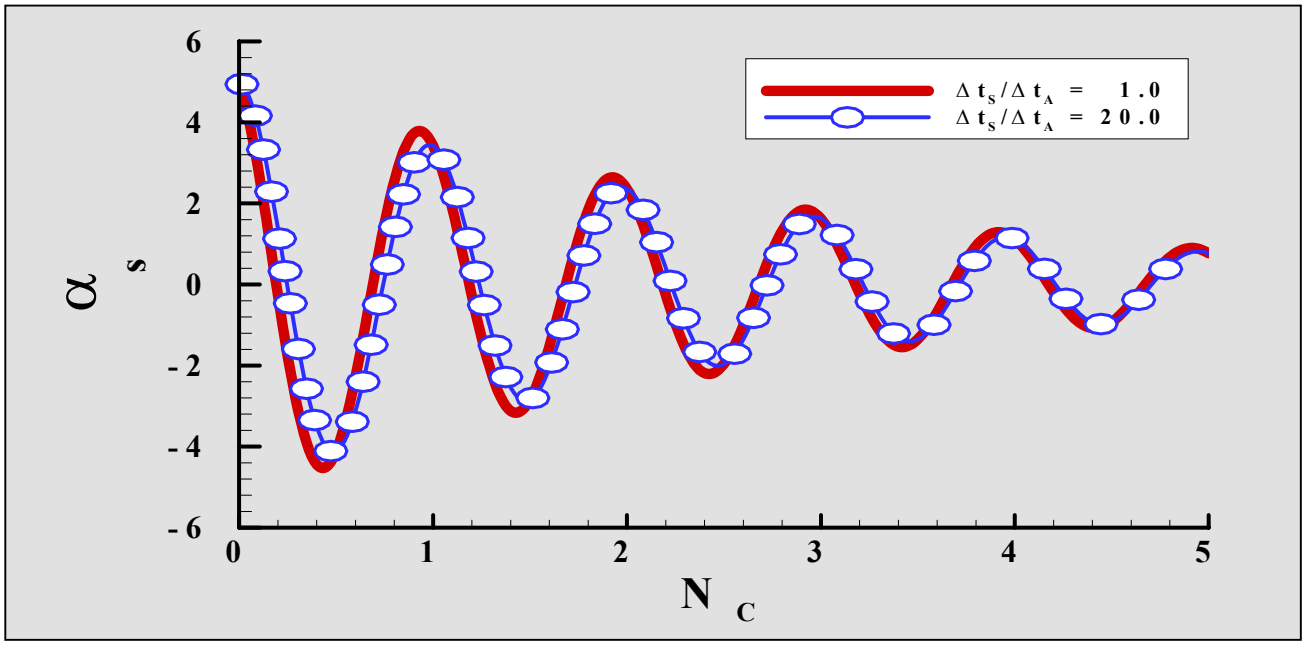

Figure 13. Effect of Varing the Time Step Differential Between the Aerodynamics and Structural Solvers in Coupled Computations.

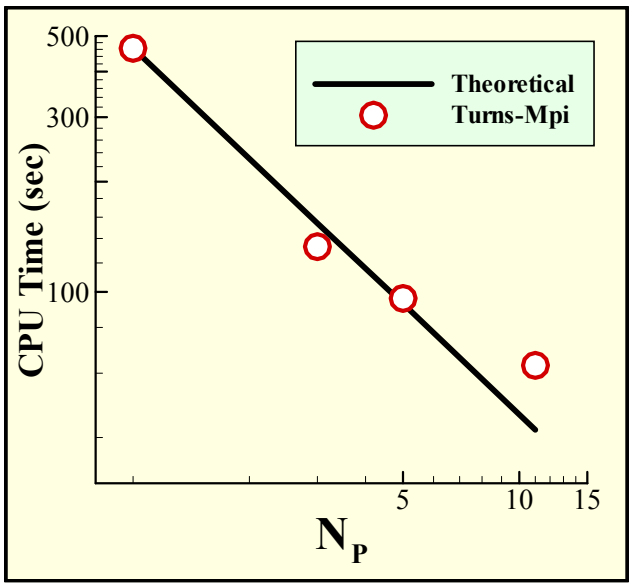

Figure 14. Scalability of TURNS_MPI for the computation of two-dimensional steady flow about a NACA 0015 airfoil.

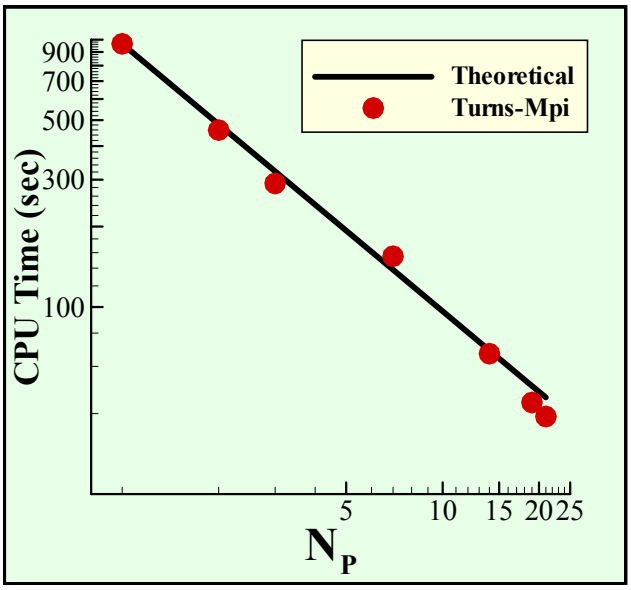

Figure 15. Scalability of TURNS_MPI for the computation of three-dimensional steady flow about an OLS blade.